

## **REVENDEICATIONS**

1. (Original) Procédé de vérification fonctionnelle d'un modèle logiciel (40) d'un circuit intégré à la demande (ASIC), en langage de bas niveau (tel que par exemple de type HDL) traitant de façon séparée l'établissement du modèle et la mise au point des tests de vérification fonctionnelle à appliquer au modèle du circuit pour constituer une plate-forme de vérification comportant les deux étapes suivantes :

- constitution d'un émulateur autonome (1) de circuit obtenu en remplaçant le modèle en langage de bas niveau (de type HDL) de description physique du circuit en projet à valider, par une description abstraite de haut niveau (par exemple C++) générant des structures de données de réponse conformes à la spécification fonctionnelle (20) du projet en fonction des stimuli reçus, ce mode étant dit « mode émission »,

intégration du modèle logiciel (40) en langage de bas niveau (de type HDL) du circuit résultant du projet dans une plate-forme de vérification, et constitution du branchement de l'émulateur autonome (1) de circuit, précédemment validée, en parallèle sur les interfaces du modèle logiciel (40) du circuit, et du branchement d'un émulateur d'environnement (11,21,22), et

- utilisation de cette plate-forme comme référence pour la validation des données de réponses émises par le modèle logiciel (40) du circuit, ce mode étant dit « mode vérification ».

2. (Original) Procédé selon la revendication 1, dans lequel :

- un utilisateur élabore, au moyen d'un système de traitement de données, la configuration de simulation autonome (1) correspondant au modèle logiciel (40) de l'ASIC au moyen de la spécification fonctionnelle (20),
- l'utilisateur écrit, à partir de la spécification fonctionnelle (20), et mémorise, dans une plate-forme de test (21, 22, 23) de modèles de circuits intégrés, un programme (51) de test du modèle (40) de l'ASIC, comportant des séquences de stimuli d'entrée à fournir au modèle logiciel (40) de l'ASIC, auxquelles la configuration de simulation autonome (1) fait correspondre, en fonction de la spécification fonctionnelle (20), des séquences de stimuli de sortie,
- l'utilisateur relie ensemble et active la configuration de simulation autonome (1) et la plate-forme de test (21, 22, 23), et
- il observe les stimuli de sortie du modèle de type HDL (40) de l'ASIC pour valider fonctionnellement l'ensemble constitué par le modèle logiciel du circuit ASIC et le programme de test de validation (210), et ainsi valider le modèle logiciel (40) par rapport à la spécification fonctionnelle (20).

3. (Currently amended) Procédé selon ~~l'une des revendications 1 et 2~~ la revendication 1, dans lequel, la configuration de simulation autonome (1) communiquant avec l'utilisateur pour commander l'activation de modèles, préalablement établis et mémorisés, de séquences de stimuli d'entrée définis dans un langage de programmation de haut niveau, et commande l'activation de programmes associés (90) de validation progressive de séquences de test déterminées à partir des modèles.

4. (Currently amended) Procédé selon ~~l'une des revendications 1 à 3~~ la revendication 1, dans lequel l'utilisateur écrit et fournit la spécification

fonctionnelle (20) dans un langage de programmation de bas niveau, spécifiant des modèles fonctionnels de circuits.

5. (Currently amended) Procédé selon ~~l'une des revendications 1 à 4~~ la revendication 1, dans lequel l'utilisateur fournit la spécification fonctionnelle (20) sous la forme d'un programme en langage de bas niveau, de modèles (de type HDL) fonctionnels de circuits, et d'un programme en langage de haut niveau, de modèles fonctionnels ((C++) symboliques de circuits), et il commande la configuration de simulation autonome (1) pour effectuer une co-simulation par synchronisation d'exécution des deux programmes de spécification.

6. (Currently amended) Procédé selon ~~l'une des revendications 1 à 4~~ la revendication 1, dans lequel la plate-forme de test vérifie que les réponses du modèle logiciel de l'ASIC sont situées dans des plages de temps de réponse spécifiées dans la spécification fonctionnelle (20).

7. (Original) Plate-forme de vérification de modèle logiciel de circuit intégré à la demande (ASIC), caractérisé par le fait qu'elle comporte des moyens de traitement de données permettant à un client de sélectionner des modèles de test engendrant des stimuli d'entrée de l'ASIC, ces moyens de traitement étant agencés pour lire des éléments (20) de spécification fonctionnelle de l'ASIC, et comportant des programmes (90) agencés pour élaborer un programme de test (51) de validation fonctionnelle constitué de stimuli de sortie à partir des stimuli d'entrée et des éléments de spécification fonctionnelle (20).

8. (Original) Plate-forme de vérification selon la revendication 7, comportant une bibliothèque de modèles fonctionnels de blocs de circuits

pour ASIC et des moyens de sélection des modèles par un fichier de définition de la configuration pour constituer un modèle correspondant à la spécification fonctionnelle de l'ASIC intégré à la définition de son environnement.

9. (Currently amended) Plate-forme de vérification selon ~~l'une des revendications 7 et 8~~ la revendication 7, dans laquelle il est prévu, dans une liaison le reliant au client, deux circuits en série d'adaptation de langage de programmation (11, 12) agencés pour transformer des commandes en langage de haut niveau (C<sup>++</sup>), utilisé par le client, en commandes en langage de bas niveau (de type HDL), exploitables par le modèle de l'ASIC, et pour, respectivement, retransformer les commandes en langage de bas niveau en commandes en langage de haut niveau.

10. (Currently amended) Plate-forme de vérification selon ~~une des revendications 7 à 9~~ la revendication 7, caractérisé par le fait qu'elle comporte des moyens (90, 10) d'exécuter ses traitements en même temps que la simulation qu'il peut interrompre dès la détection d'une erreur au moment même de l'apparition de l'erreur.

11. (Currently amended) Plate-forme de vérification selon ~~une des revendications 7 à 10~~ la revendication 7 caractérisé par le fait que les éléments (20) de spécification fonctionnelle sont constitués d'une table de vérité, ou de comportement, correspondant aux fonctions des diverses parties ou divers éléments de circuit fonctionnels du modèle logiciel (40) de l'ASIC, et des plages de retard de propagation à respecter entre chaque entrée et chaque sortie.

12. (Currently amended) Plate-forme de vérification selon ~~une des revendications 7 à 11~~ la revendication 7, caractérisé par le fait qu'elle dispose d'une antémémoire (962) pour mémoriser les blocs utilisés par les nœuds d'après leur adresse et des moyens de gérer, pour une adresse utilisée par un ou plusieurs nœuds, un vecteur de présence avec un témoin de présence par nœud.

13. (Currently amended) Plate-forme de vérification selon ~~une des revendications 7 à 12~~ la revendication 7, caractérisé par le fait que les programmes (90) sont orientés objet et l'émulateur est structuré en un ensemble de classes permettant de gérer une collection d'hypothèses d'exécution d'une transaction sur un bloc mémoire du modèle logiciel, et de gérer également les transactions en collisions, c'est-à-dire utilisant concurremment un même bloc mémoire.

14. (Currently amended) Plate-forme de vérification selon ~~une des revendications 7 à 13~~ la revendication 7 caractérisé par le fait que les algorithmes des programmes (90) de l'émulateur réalisent les fonctions suivantes : génération des prédictions, élimination des prédictions, ré-ajustement de mauvaise prédiction, réduction du nombre d'hypothèses valides et terminaison de collisions.

15. (Currently amended) Plate-forme de vérification selon ~~une des revendications 7 à 14~~ la revendication 7 caractérisé par le fait qu'elle est utilisée en émulateur de circuit routeur, de circuit à antémémoire ou de circuit routeur à antémémoire.

16. (Currently amended) Plate-forme de vérification, selon ~~une des revendications 7 à 15~~ la revendication 7, pour tester un modèle logiciel de

circuit intégré à la demande (ASIC), caractérisée en ce qu'elle comporte un émulateur d'ASIC (1) pour commander un comparateur (23) prévu pour recevoir des valeurs générées par le modèle logiciel de circuit ASIC testé, sur réception de stimuli envoyés par au moins un circuit (21) générateur de stimuli, mémorisant le programme de test, une interface (11) de traduction des stimuli d'un langage élaboré vers un langage de bas niveau correspondant à celui du modèle logiciel, et des moyens de validation de la vérification en cas de détection de collision par le comparateur (23).

17. (Currently amended) Plate-forme de vérification selon ~~une des revendications 7 à 16~~ la revendication 7, caractérisée en ce que les moyens de sélection de la réponse à des stimuli dépendants de la constitution des circuits testés sont constitués d'un modèle élaboré grâce à des moyens de sélection, parmi une bibliothèque, de modèles fonctionnels associant, à chacun des modèles, les réponses à un stimulus donné, le modèle correspondant à la constitution du circuit à tester.

18. (Currently amended) Plate-forme de vérification selon ~~une des revendications 7 à 17~~ la revendication 7, caractérisée en ce qu'elle comporte des moyens (7) de mémorisation des réponses ainsi sélectionnées pour constituer un modèle de test (70) à appliquer au circuit testé lors de la réception de stimuli.

19. (Currently amended) Plate-forme de vérification selon ~~une des revendications 7 à 18~~ la revendication 7, caractérisée en ce que chaque transaction est constituée, au niveau de chaque interface, d'un paquet requête et d'un ou plusieurs paquets réponses associés, dont les valeurs des paramètres et/ou les contraintes temporelles d'émission des paquets peuvent

être forcées à partir du programme de test fonctionnel exécuté par l'émulateur de l'environnement, qui traduit de façon adéquate l'ensemble de ces paramètres lors de l'émission des paquets aux bornes du modèle logiciel du projet.

20. (Original) Plate-forme de vérification selon la revendication 14, caractérisée en ce que la génération des prédictions est effectuée par l'émulateur du circuit sans avoir à prélever d'informations supplémentaires sur le fonctionnement interne du circuit en projet.